平成30年度 茨城大学工学部情報工学科
卒業研究論文

文の持つ情報量を用いた
ニューラル機械翻訳の訳抜け検出

平成31年2月5日

茨城大学 工学部 情報工学科
15T4057F 藤井 真
指導教員: 新納 浩幸 教授

平成 30 年度 茨城大学工学部情報工学科卒業研究

文の持つ情報量を用いた ニューラル機械翻訳の訳抜け検出

氏名：15T4057F 藤井 真
指導教員：新納 浩幸 教授

論文要旨

本論文では, ニューラル機械翻訳 (Neural Machine Translation;NMT) の訳抜け検出に対して原文の情報量ベースの手法を提案する.

近年の機械翻訳はニューラルネットワークの技術を利用することで人が受け入れやすい自然な文章をもたらしている. しかし, NMT はニューラルネットワークを用いて訳文を生成するため, 従来の機械翻訳手法である統計的機械翻訳 (Statistical Machine Translation;SMT) で問題となりにくかった「訳抜け」を起こしやすい. 訳抜けとは原文に存在していた文意を訳文において意味的, 単語的に欠いてしまう現象である.

NMT の訳抜けした内容の検出に関しては, ニューラルネットワークの累積アテンション確率と逆翻訳による文生成の確率を利用した Goto らの手法がある. これはニューラルネットワーク内で出力までの評価に用いられるアテンション確率の高低で翻訳された内容か否かを判断する手法と, 原文から NMT によって出力した訳文をもう一度 NMT によって原文言語に戻し, 原文言語同士の比較によって訳抜けした内容を検出する手法である. ニューラルネットワーク内部のアテンション確率を参照する必要がある点や二重の翻訳を行う点等, 処理コストが高い.

本研究は訳抜けした内容の検出ではなく, 訳抜けの有無の検出を対象にしている. 検出の方法は原文中の単語が存在する確率を元に, その単語が持つ情報量を設定する. 文はそれら単語の集合のため, 文全体のあるべき情報量が仮定できる. 同様に仮定した訳文の情報量と比較することで訳抜けを検出するものである.

本手法の特徴は, 逆翻訳や NMT の内部へ干渉しないため低コストに検出を行える点である. また, 入力として与える情報が原文と機械翻訳文の対のみという点である. 本研究は NMT のモデルを対象としているが, 理論上は SMT のモデルや人が訳した文であっても訳抜けの検出を行える.

実験では英語のニュースの文を WEB ページから取得し原文とした. その原文を NMT により訳文とし, それら原文と訳文から提案手法によって訳抜け状態の文を検出する.

現状の結果として, 訳抜けの検出精度は検出量とのトレードオフの状態となる. 全体の 2-3%にあたる訳抜け検出においては高精度な結果を得られた.

目 次

第1章　はじめに

1.1　概要

本論文では, ニューラル機械翻訳 (Neural Machine Translation;NMT) の訳抜け検出に対して原文の情報量ベースの手法を提案する.

近年の機械翻訳はニューラルネットワークの技術を利用することで人が受け入れやすい自然な文章をもたらしている. しかし, NMT はニューラルネットワークを用いて訳文を生成するため, 従来の機械翻訳手法である統計的機械翻訳 (Statistical Machine Translation;SMT) で問題となりにくかった「訳抜け」を起こしやすい. 訳抜けとは原文に存在していた文意を訳文において意味的, 単語的に欠いてしまう現象である.

NMT の訳抜けした内容の検出に関しては, ニューラルネットワークの累積アテンション確率と逆翻訳による文生成の確率を利用した Goto らの手法 [1] がある. これはニューラルネットワーク内で出力までの評価に用いられるアテンション確率の高低で翻訳された内容か否かを判断する手法と, 原文から NMT によって出力した訳文をもう一度 NMT によって原文言語に戻し, 原文言語同士の比較によって訳抜けした内容を検出する手法である. ニューラルネットワーク内部のアテンション確率を参照する必要がある点や二重の翻訳を行う点等, 処理コストが高い.

本研究は訳抜けした内容の検出ではなく, 訳抜けの有無の検出を対象にしている. 検出の方法は原文中の単語が存在する確率を元に, その単語が持つ情報量を設定する. 文はそれら単語の集合のため, 文全体のあるべき情報量が仮定できる. 同様に仮定した訳文の情報量と比較することで訳抜けを検出するものである.

本手法の特徴は, 逆翻訳や NMT の内部へ干渉しないため低コストに検出を行える点である. また, 入力として与える情報が原文と機械翻訳文の対のみという点である. 本研究は NMT のモデルを対象としているが, 理論上は SMT のモデルや人が訳した文であっても訳抜けの検出を行える.

実験では英語のニュースの文を WEB ページから取得し原文とした. その原文を NMT により訳文とし, それら原文と訳文から提案手法によって訳抜け状態の文を検出する.

現状の結果として, 訳抜けの検出精度は検出量とのトレードオフの状態となる. 全体の 2-3%にあたる訳抜け検出においては高精度な結果を得られた [4].

1.2 構成

本論文では, 文の持つ情報量を用いたNMTの訳抜け検出について示すため, 2章でNMT関連知識である機械翻訳, SMT, 数学モデルとしてのニューラルネットワーク, NMTのエンコーダー・アテンション機構・デコーダー, 訳抜けについて順次説明する. 3章は用いたコーパスの収集方法と様相を説明する。4章では訳抜けを検出するために提案する手法の内容, 5章ではその手法を用いた実験の内容と結果を示す.

また本論文のプログラミング等実行環境は特記のない場合, 以下の表1.1のとおりである.

表 1.1: 実行環境

OS	Ubuntu 16.04.5 LTS
Python	Python 3.6.4

第2章　ニューラル機械翻訳

2.1　機械翻訳

自然言語処理の分野における機械翻訳とは, 自然言語つまり人間が日常的に用いる言語を別の自然言語に変換するタスクを機械的に処理するものである. その発展には自然言語処理技術や計算機性能の向上等に起因して様々な段階を経ている. 黎明期においては人手で整備されたルールに基づく機械翻訳, 後に過去の翻訳用例 (対訳コーパス) を組み合わせて翻訳を行う手法が提案された. これは用例に基づく翻訳 (Example-Based Machine Translation; EBMT) と呼ばれる. 1980 年代後半に IBM の研究グループが SMT について研究を開始した. これは NMT が主流となる近年まで用いられてきた手法である. この拡張には 2003 年に提案された句に基づく翻訳 (Phrase-Based SMT; PBSMT) がある. その後, 計算機性能の向上により大規模計算モデルのニューラルネットワークが一般的になった. ニューラルネットワークを機械翻訳に取り入れたものが NMT である. 2016 年 11 月に Google 翻訳が NMT を採用し成果を残している.

2.2 SMT

統計的機械翻訳のこと. 統計的手法を用いて原文言語から訳文言語に変換する. ニューラルネットワークを機械翻訳が取り入れる以前は PBSMT が主流であった. PBSMT では原文と訳文の単語対応を統計的に自動推定し, その自動推定結果からフレーズテーブルを構築する. このフレーズテーブルは翻訳表現の集合であり機械翻訳の表現力につながる. 言語ごとのモデル処理や翻訳の際の重み付け等, さまざまな段階を経て翻訳精度を高めている. 各段階においてはそれぞれ独立したツールを用いることが多い. 翻訳時の挙動の概要を図 2.1 に示す. まず単語化を行い, 次に句を考えてフレーズテーブルから訳出候補を選出する. その後, 言語ごとの自然さを元に重み付けを行い翻訳候補を選定していく. 原文の部分的置き換えが統計的に不都合なく全体に及んだ時点で翻訳終了となる.

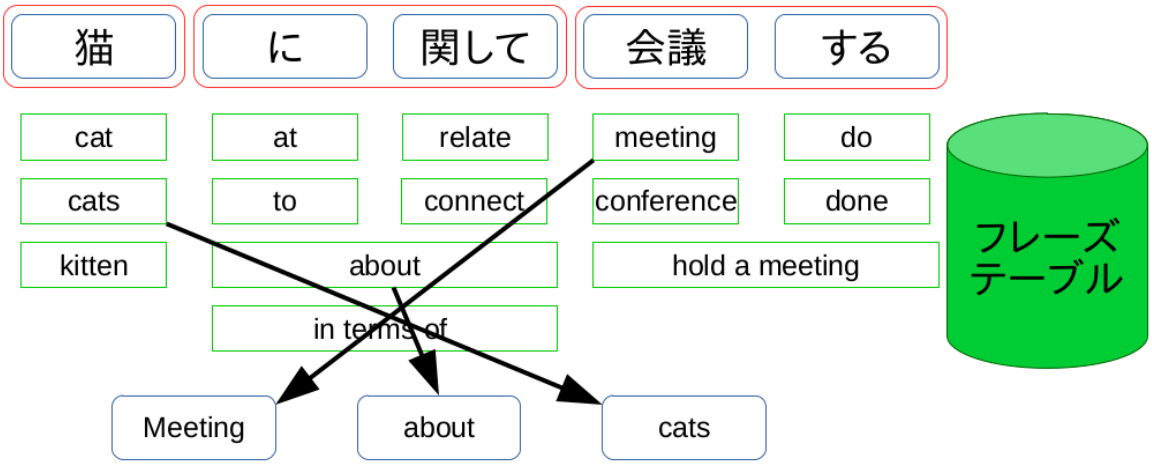

図 2.1: PBSMT における翻訳処理概要

2.3 NMT

ニューラル機械翻訳のこと. 大規模な翻訳用例をニューラルネットワークの手法によって学習し翻訳モデルを構築したもの. SMT は翻訳処理の様々な段階で, より適性のあるツールを組み合わせてモデル化しているが, NMT では入力から出力まで単一のモデルで完結することが多い. 主な構成要素は 3 つで, 1 つ目はエンコーダー (encoder) である. これは入力された原文言語の文を数理モデルで扱うために実数値の集合であるベクトルとして表現する役割である. 2 つ目はアテンション機構である. ベクトル化された入力文を出力文とするためにどの部分が重要かを判断する. 3 つ目はデコーダー (decoder) である. 数理モデルで扱ったベクトルを訳文言語としてどのように出力するかを入力文とアテンション機構を元に判断し, 訳文として出力する.

2.3.1 ニューラルネットワーク

ここでは数理モデルとしてのニューラルネットワークを説明する. 人間の脳には神経細胞ニューロンが存在し, 情報処理や情報伝達を行っている. その機構を模した数理モデルである. 基本となる処理は多層化した線形関数であり, その各所に活性化関数と呼ばれる非線形関数を通すことで単なる合成関数からいくらかの揺れをもたせている. この処理によってもたらされる出力と理想の出力の誤差を最小とするように各線形関数の重みを学習していく.

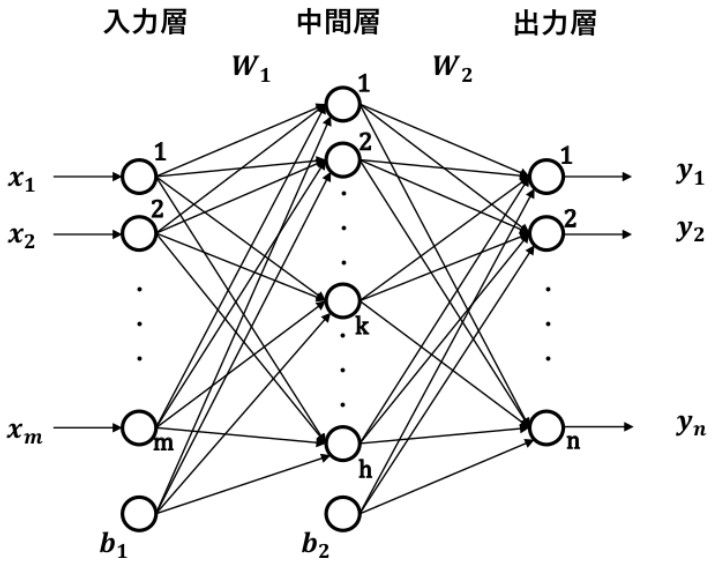

図 2.2: ニューラルネットワークの概要図

2.3.2 エンコーダー

自然言語を数理モデルで扱うための前処理を行う. 翻訳に用いられる言語についてそれらの各単語を分散表現と呼ばれる数百次元の実数値ベクトルに変換していく. この操作は言葉を数値に埋め込むことから, エンベディング (embedding) と呼ばれる. この単語ベクトルは単語の意味を数値化したものと捉えることができ, 言葉の意味上の足し引き等を行える状態となる.

2.3.3 アテンション機構

エンコーダーによってもたらされたベクトルとデコーダーの内部状態を参照し, 次にどの単語を訳出すべきか判断する情報を蓄える. 基本的には入力文全ての単語について確率的に重要度を判定する. 1 単語訳出した際に再度この確率 (アテンション) を計算して次に重要と考える点を移していく.

2.3.4 デコーダー

数理モデルで扱われていた自然言語の後処理を行う. 出力されるべき分散表現の値に沿った単語を準備し自然言語化する. その入力にはアテンション機構の情報と一つ前に出力した単語を用い, 次にどの単語を訳出すべきかを決定し順次訳出する. つまり, NMT の訳文生成は先頭の文字列から順に訳出されていき, 文末にあたる EOS(End of Sentence) が確率上出力されるまで繰り返される.

2.4 訳抜けの問題

訳抜けとは, 原文に存在していた文意を訳文において意味的, 単語的に欠いてしまう現象である. SMT では原文の部分的置き換えが統計的に不都合なく全体に及んだ時点で翻訳終了となるため, 訳抜けのある状態で翻訳を終了することは少ない. しかし, NMT はその訳文を先頭から順次, 学習したモデルの確率によって出力していくため未知語や特殊表現等によりモデルの学習が不足していた場合等, 部分的に確率を低く設定し訳出しきれずに翻訳を終了してしまうことがある. 訳抜けの規模としては, 1 単語から句, 文の半分以上を占める節まで抜ける場合がある. 表 2.1 は訳抜けを起こした NMT の例である. 人物の所属国の情報や状況の説明が一部抜けて訳されている.

表 2.1: 訳抜けの例

原文	”What we feared has happened,” Belgian Prime Minister Charles Michel said, as he appealed for calm.
機械翻訳文	チャールズ・ミシェル首相は, 「我々が恐れていたことが起こった」と述べた.
人の翻訳例	「恐れていたことが起こってしまった」と ベルギーの シャルル・ミシェル首相は 事態の沈静化を求めながら 述べた.

第3章　対訳コーパス

本研究で用いた対訳コーパスについて収集方法とデータの様相を説明する. 提案手法では翻訳の原文となる英文のみを集めればよいが, 訳抜けの判定や今後の改良を目的として, 初期の段階で対象英文を人間が翻訳した場合の文もあわせて収集している.

3.1　収集方法

提案手法に必要な英文について 3,000 文を目安に WEB ページから取得している. その英文を Google の NMT(以下「GNMT」という) を用いて翻訳している.

3.1.1　BeautifulSoup

python のライブラリ. HTML や XML ファイルから任意の情報を取得する等に用いる. 本研究では WEB ページ内の英ニュース文を取得するために用いた. 以下に図 3.1 として今回対象にした英ニュース文を含む HTML ファイルの一部を示す. 図 3.2 としてターミナル上での BeautifulSoup のインストールの方法, 図 3.3 として python によるの BeautifulSoup の実行の様子を示す.

```
...
    <p class="PrinterFriendlyP">The Niigata Prefectural Government
      completed the culling of 540,000 chickens at two <span class
      ="nounlink"><a style="cursor:pointer;" onclick="showChuPopup(
      event,'<b>poultry farms</b><br />養鶏場');return false;"
      onmouseover="showChuPopup(event,'<b>poultry farms</b><br />養
      鶏場');" onmouseout="endChuPopup();">poultry farms</a></span>
       Dec. 4 to <span class="nounlink"><a style="cursor:pointer;"
      onclick="showChuPopup(event,'<b>contain</b><br />～を封じ込め
      る');return false;" onmouseover="showChuPopup(event,'<b>
      contain</b><br />～を封じ込める');" onmouseout="endChuPopup
      ();">contain</a></span> an out-break of the highly <span
      class="nounlink"><a style="cursor:pointer;" onclick="
      showChuPopup(event,'<b>virulent</b><br />感染力の強い');
      return false;" onmouseover="showChuPopup(event,'<b>virulent</
      b><br />感染力の強い');" onmouseout="endChuPopup();">virulent
      </a></span> H5 <span class="nounlink"><a style="cursor:
      pointer;" onclick="showChuPopup(event,'<b>strain</b><br />
      型');return false;" onmouseover="showChuPopup(event,'<b>
      strain</b><br />型');" onmouseout="endChuPopup();">strain</a
      ></span> of bird flu.</p>

    <p class="PrinterFriendlyP">About 230,000 chickens were killed
      at a farm in Joetsu, following the completion of a similar
      procedure to cull 310,000 chickens in the village of Sekikawa
       days earlier.</p>

    <p class="PrinterFriendlyP">In nearby Aomori Prefecture, 4,720
      ducks were culled Dec. 3 at a farm after it was found
      infected by the H5 strain. (Kyodo)</p>
...
```

図 3.1: 英ニュース文を含んだ HTML ファイルの一部

```
$ pip install bs4
```

図 3.2: ターミナル上での BeautifulSoup のインストールの方法

```
>>> from urllib import request
>>> from bs4 import BeautifulSoup
>>> targetURL = "http://st.japantimes.co.jp/news/?p=nm20161216"
>>> from urllib import request
>>> from bs4 import BeautifulSoup
>>> html = request.urlopen(targetURL)
>>> soup = BeautifulSoup(html, "html.parser")
>>> ens = soup.find_all("p", class_="PrinterFriendlyP")
>>> for en in ens:
...     print(en.text)
...
The Niigata Prefectural Government completed the culling of 540,000
   chickens at two poultry farms Dec. 4 to contain an out-break of
  the highly virulent H5 strain of bird flu.
About 230,000 chickens were killed at a farm in Joetsu, following
  the completion of a similar procedure to cull 310,000 chickens in
   the village of Sekikawa days earlier.
In nearby Aomori Prefecture, 4,720 ducks were culled Dec. 3 at a
  farm after it was found infected by the H5 strain. (Kyodo)
```

図 3.3: python によるの BeautifulSoup の実行の様子

3.1.2 NMT による訳文の出力方法

本研究では英文 3,000 文について,GNMT を用いて翻訳している. 翻訳の方法は python により 3,000 文の英文を載せた html ファイルを作成し, Google の WEB ブラウザ Chrome に読み込ませて, 英文ページの自動翻訳機能を用いて翻訳を行っている. Google が提供する通常の翻訳専用の WEB ページと同様の翻訳が行われていることを確認している.

3.2 コーパスの様相

コーパスとして用いている英文 3,000 文のファイル news3000E.txt と機械翻訳後のファイル news3000T.txt の様相をそれぞれ図 3.4 と図 3.5 として以下に示す.

```
1   The former Miss International, Ikumi Yoshimatsu, filed criminal
  and civil charges Dec. 11 against Genichi Taniguchi, one of
 Japan's most powerful talent agencies' executives, for stalking
 her and attempting to ruin her career.
2   The harassment began after she refused to sign up with a talent
  agency that was long rumored to have gang connections.
3   Yoshimatsu has provided tape recordings, videos and photographs
  detailing the stalking.

...

2998    The crisis is being blamed on habitat loss from housing
 development.
2999    This has led to koalas getting killed by cars and trucks,
 as well as stress-related diseases.
3000    The 34-million-dollar program in the southeastern state of
 New South Wales sets aside almost 24,000 hectares of public
 forest as a koala reserve.
```

図 3.4: news3000E.txt

```
1   元ミスインターナショナル，吉松郁美氏は12月11日，日本を代表する
 有能な人材派遣会社の1人である谷口元一氏に対して，彼女を侮辱し，キ
 ャリアを破滅させようとした刑事および民事訴訟を起こした．
2   彼女がギャングとのつながりがあると長い間噂されていた才能のある
 機関への申し込みを拒否した後，嫌がらせが始まりました．
3   吉松はストーキングの詳細を記録したテープレコーディング，ビデオ，
  写真を提供した．

...

2998    危機は住宅開発による生息地の喪失が原因であると非難されてい
 ます．
2999    これは，コアラが車やトラック，そしてストレス関連の病気によ
 って殺されることにつながりました．
3000    ニューサウスウェールズ州南東部の州での3400万ドルのプログラ
 ムは，コアラ保護区として約24,000ヘクタールの公有林を確保していま
 す．
```

図 3.5: news3000T.txt

第4章　提案手法

本研究は原文の単語が持つ情報量を元に文全体のあるべき情報量を仮定し, 同様に仮定した訳文の情報量と比較することで訳抜けを検出するものである. 図 4.1 は提案手法の概要図である. 原文から訳文を生成し, 各々を自立語化し情報量を比較している.

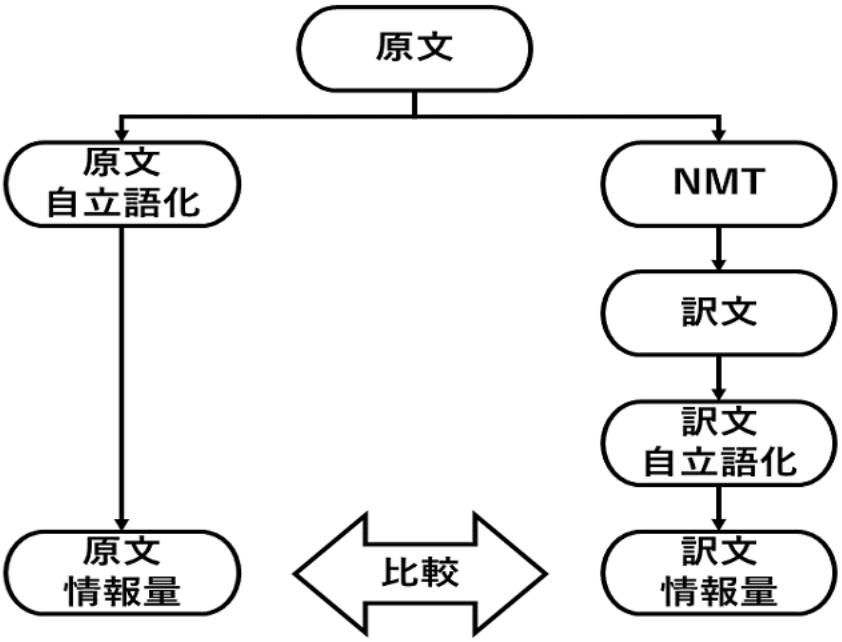

図 4.1: 提案手法の概要図

4.1　単語化

原文, 訳文ともに形態素解析を行って単語化する. 形態素解析とは文法構造等の情報が無いテキストデータについて, その言語の文法構造情報を付与するタスクである. 機械翻訳後の日本語文には Mecab を, 原文の英語文には nltk を用いた. どちらも python のパッケージである.

4.1.1 Mecab

Mecab は京都大学情報学研究科—日本電信電話株式会社コミュニケーション科学基礎研究所共同研究ユニットプロジェクトを通じて開発されたオープンソース形態素解析エンジンである. 以下にターミナル上での Mecab のインストールの方法と python によって Mecab を用いた例をそれぞれ図 4.2 と図 4.3 示す.

```
$ pip install mecab-python3
```

図 4.2: ターミナル上での Mecab のインストールの方法

```
>>> import MeCab
>>> tagger = MeCab.Tagger()
>>> print(tagger.parse("古池や蛙飛び込む水の音"))
古池　　名詞,固有名詞,地域,一般,*,*,古池,フルイケ,フルイケ
や　助詞,並立助詞,*,*,*,*,や,ヤ,ヤ
蛙　名詞,一般,*,*,*,*,蛙,カエル,カエル
飛び込む　　動詞,自立,*,*,五段・マ行,基本形,飛び込む,トビコム,トビ
  コム
水　名詞,一般,*,*,*,*,水,ミズ,ミズ
の　助詞,連体化,*,*,*,*,の,ノ,ノ
音　名詞,一般,*,*,*,*,音,オト,オト
EOS
```

図 4.3: python によって Mecab を用いた例

4.1.2 nltk

nltk は Natural Language Toolkit の略である. python のオープンソースパッケージであり, 英語を中心として多方面の言語処理をサポートしている. 以下にターミナル上での nltk のインストールの方法と python によって nltk を用いた例をそれぞれ図 4.4 と図 4.5 示す.

```
$ pip install nltk
```

図 4.4: ターミナル上での nltk のインストールの方法

```
>>> import nltk
>>> tokens = nltk.word_tokenize("Lack of sleep is becoming an
  increasingly serious problem in Japan.")
>>> for token, pos in nltk.pos_tag(tokens):
...     print(token, pos)
...
Lack NN
of IN
sleep NN
is VBZ
becoming VBG
an DT
increasingly RB
serious JJ
problem NN
in IN
Japan NNP
. .
```

図 4.5: python によって nltk を用いた例

単語の横に出力されている NN や IN 等が品詞を表している.

4.2 文の情報量

文を構成する単語の中で情報を持つと考えられる自立語を元に算出する. 自立語 w が言語においてどの程度情報を持っているかという度合いを自立語 w の言語における存在確率から設定する. 文としての情報量 S は自立語 w を組み合わせたものとして以下の式で定める.

$$S = \sum_{i=1}^{K} \log_2 \left(\frac{N}{w_{ni}} \right) \tag{4.1}$$

ここで N は言語の総自立語数である. w_n は言語中のそれぞれの自立語の出現回数であり, i は S 文中に現れる K 個の自立語の出現順である.

4.3 訳抜けの判定

訳抜けの判定は人手によって行った. 訳抜けの基準は原文から得られるべき情報が抜けているか否かを基準としている. 誤訳は原文の内容を訳出しているので訳抜けとしていない.

式 4.1 の値は単語数が多い文や稀な単語が使われるほど高くなる傾向がある. 単に原文の情報量 O と訳文の情報量 T の差をとった場合, 訳抜けが少ない文であっても長文であることを理由に, 割合として情報量の差が大きく出てしまうことがある.

そのため, 訳抜けの指標 L としては $L = \frac{O-T}{O}$ として原文の情報量を元にした情報損失割合を考える.

第5章　実験

5.1　実験設定

機械翻訳は英語の原文から日本語への訳文を対象とする. 機械翻訳には GNMT を用いた. ニューラルネットワークの学習の性質上, または Google の NMT モデル運用上の理由からか翻訳結果が一様に得られない場合がある. 本研究は, 2019 年 1 月 4 日における GNMT の結果を元にしている.

実験に用いる原文の内容は, 英語のニュース文 3,000 文を対象とする. 文量による効果をみるために, 500 文単位で提案手法を用いた実験も行った. 図 5.1 は今回用いたコーパスの 1 文あたりの規模を日本語の自立語数で表し, その分布を表したものである. 平均 16.3, 中央値 16 の自立語を含む文を対象にしている.

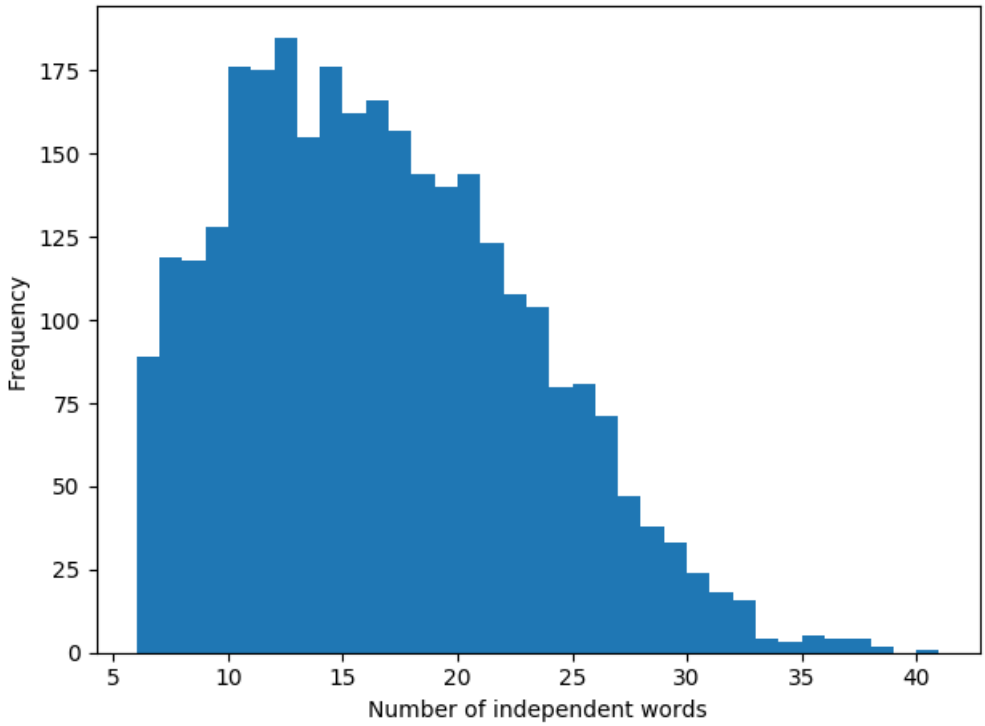

図 5.1: 訳文における自立語数分布

情報として扱う自立語について, 訳文となる日本語は自立語の概念があるため Mecab で形態素解析した品詞の助詞・助動詞を抜いたものを対象とする. 原文である英語は, 日本語

の自立語に対応するものを nltk の品詞分類の中から選択している.

表 5.1 は, これらの設定で 3,000 文の自立語がどの程度になるかを示したものである.

表 5.1: 3,000 文の自立語数

	総数	種類数
英語原文	59,145	11,384
機械訳文	48,774	9,954

評価には真陽性率を用いる. ここで陽性とは訳抜けありのことである. 提案手法によって検出された陽性を分母に, その中から真に陽性なものを分子にしたものが真陽性率となる.

5.2 実験結果

5.2.1 情報損失割合による検出量と精度

図 5.2 は提案手法を用いて, 英ニュース文の訳抜け検出を行った結果である. 情報損失割合を高めると検出量が減り, 精度が高まる. 全体の 2-3%にあたる検出であれば約 90%以上の精度で検出している.

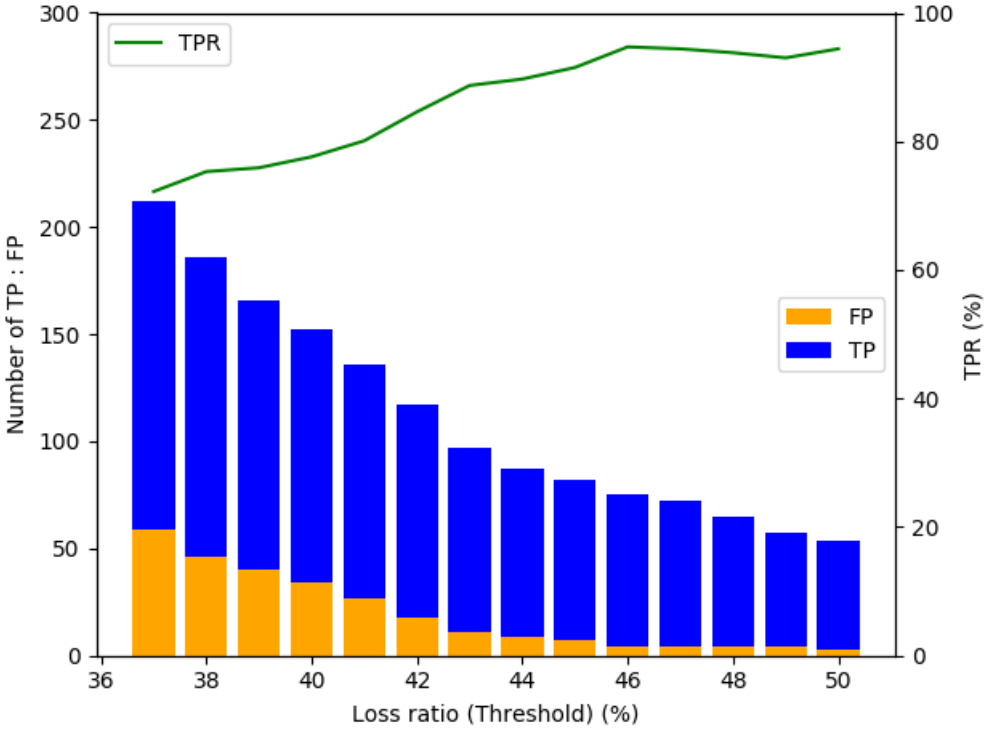

図 5.2: 提案手法による検出量と精度

5.2.2 NMT による文の情報量変化と訳抜けの例

表 5.2 は本手法によって情報が失われた, または増加したと判断された文の例である.

表 5.2: 情報量の差が顕著な例

	情報量	文
原文 1	280	”We should all do everything we can to see that, in Jo’s memory, we bring an end to the acceptance of loneliness for good,” Prime Minister Theresa May said in a statement.
機械翻訳	57	テレサ・メイ首相は声明で次のように述べている.
原文 2	403	The Russian ambassador to Turkey was shot in the back and killed at an Ankara art gallery Dec. 19 by an off-duty police officer who shouted ‘’Don ’t forget Aleppo ” and “ Allahu Akbar ” as he opened fire.
機械翻訳	177	トルコへのロシア大使は 12 月 19 日, アンカラ美術館で 「アレッポを忘れないで」と「アッラーフ・アクバル」を 発砲して怒鳴ったため殺害された.
原文 3	100	Beijing has poured billions into such ambitious scientific projects.
機械翻訳	151	北京はこのような野心的な科学プロジェクトに 何十億ドルもの資金を注いできました.

原文 1 は主要内容である発言部が全て訳抜けている. 原文 2 は訳抜けの辻褄合わせに, 撃たれた人物を撃った人物とする誤訳が起きている. どちらも情報量の差が大きい。網掛け部分の英文が訳されておらず, 明らかな訳抜けを検出している. 原文 3 は本手法によって, 情報が増加したケースである. 「billions」という 1 自立語に対し, 形態素解析で「何」, 「十」, 「億」, 「ドル」と分けてしまっている.

5.2.3　コーパスの文量による提案手法の効果

文量による効果をみるために 500 文ごとに提案手法を用いた. それぞれ 3,000 文からランダムにサンプリングし, 提案手法を 100 回行った結果の平均値である. 情報損失割合の閾値は 50%で固定している. 結果は図 5.3 である.

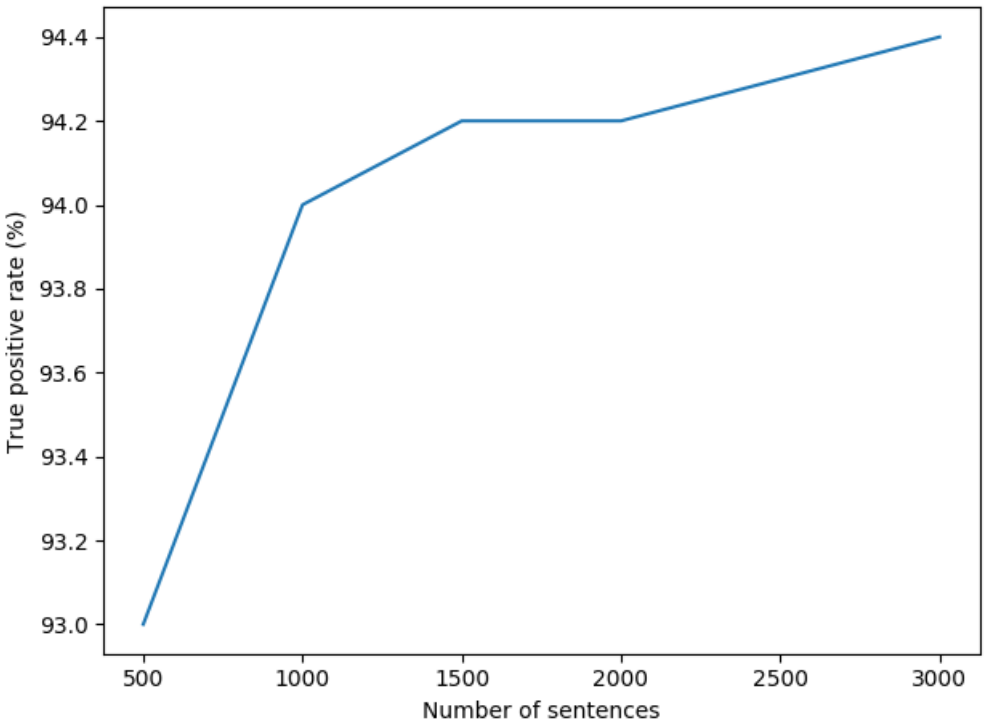

図 5.3: 提案手法による真陽性率

第6章　考察

英文 3,000 文を対象とした訳抜け検出において, 提案手法の精度は検出量とのトレードオフの状態となる. 全体の 2-3%にあたる訳抜け検出においては高精度な結果を得られた. 検出閾値 50%以上において偽陽性となったものは陽性 54 文中に 3 文あった. 主な要因は, 原文が短文であることである. 偽陽性の 3 文はそれぞれ自立語数が 7,10,11 の文であった. NMT は修飾の少ない短文における翻訳精度が高い. これは短文におけるエントロピーは低く, 訓練された領域から外れにくいためであると推測される.

また, 提案手法は自立語の選定に起因して短文における情報損失精度が粗くなることがある. これは形態素解析によって得られる日本語の自立語概念と, 英語の自立語の選出に言語間レベルで差異があるためである. 他には, New York 等の複単語表現ひいては固有表現抽出の問題が挙げられる. あるエンティティを 2 言語間で 1 対 1 に対応付けるとき, 高い形態素解析精度が必要であり単語レベルを基本とする今回の手法ではこの問題への柔軟性が低い. これらの要因から短文における精度不足が生まれると考えられる.

第7章　おわりに

本論文ではニューラル機械翻訳の訳抜け検出に対して, 原文の情報量ベースの手法を提案した. 原文の単語が持つ情報量を元に文全体のあるべき情報量を仮定し, 訳文の情報量と比較することで訳抜けを検出している. 実験では計算コスト低く, 明らかな訳抜けを検出している.

提案手法の改良としてまず挙げられる点は複単語表現と固有表現抽出である. 現在は 1 エンティティの内容を日本語 1, 英語 5 として自立語数をカウントするケースがある. 複単語表現を 1 エンティティとして認識する枠組みとして ngram 等を採用し, 固有表現抽出によって 1 エンティティ性を高めることで, 提案手法の精度向上が期待できる. 同時に提案手法の検出閾値を下げて実験を行うために, 人手による訳抜け判定作業を進め閾値が低い場合の知見も取り入れたい.

また, 本研究は計算コストを軽い制約としているためニューラルネットワークを用いていないが, 情報を持つ単語や品詞の選定についてはニューラルネットワークによる事前調査傾向を用いて重み等を新たに設定することで手法の改良を試みたい. これと形態素解析のより詳細な結果を併用する方向で研究を進めたい.

近年, 機械翻訳は人が受け入れやすい自然な文をもたらしている. しかし, 実験結果表 5.2 の原文 2 に顕著であるが, 自然な文とするための流れを重視するために事実の主客を入れ替えてしまうケース等も見受けられる. 人にとっての自然さを見せ始めたニューラルネットワークを人が理解し続ける必要があることを改めて示唆する結果となった.

謝辞

本研究を進めるにあたり, 熱心にご指導頂いた情報工学科の新納教授に深い感謝の意を表します. また, 多くのご意見ご指摘を頂きました自然言語処理研究室の皆様に心より感謝申し上げます.

参考文献

[1] Isao Goto and Hideki Tanaka. Detecting untranslated content for neural machine translation. In Proceedings of the 1st Workshop on Neural Machine Translation (ACL-2017), pp. 47-55, 2017.

[2] Ryuichiro Kimura, Zi Long, Takehito Utsuro, Tomoharu Mitsuhashi and Mikio Yamamoto. Effect on reducing untranslated content by neural machine translation with a large vocabulary of technical terms. In Proceedings of the 7th Workshop on Patent and Scientific Literature Translation, pp. 13-24, 2017.

[3] 中澤敏明. 機械翻訳の新しいパラダイム：ニューラル機械翻訳の原理. 情報管理, Vol.60, No.5, pp. 299-306, 2017.

[4] 藤井真, 新納浩幸, 古宮嘉那子. 文の持つ情報量を用いたニューラル機械翻訳の訳抜け検出. 言語処理学会第 25 回年次大会 (NLP2019), p. to appear, 2019.